ESAIN Kevin
ANTIOME Marc

Projet de construction et parcours d'un labyrinthe

jean-philippe.dubernard@univ-rouen.fr

Sommaire:

-Cheminement du raisonnement

-Manuel d'utilisation du programme

-Exemples d'éxécution

-Code

Explication:

La problématique de ce projet consiste en l'utilisation des modules et l'utilisation de types imposés pour créer un labyrinthe, l'afficher puis afficher son parcours vers la sortie.

Demarche:

Nous avons décidé pour ce projet de nous pencher sur les modules, les "Map" et les "Set", afin de comprendre le sujet en cherchant nos recherches se dirigèrent vers l'aide ocaml fournie sur le site de l'INRIA et nos questions se tournèrent vers nos professeurs de programmation fonctionnelle. Après un long moment nécessaire à notre compréhension, nous avons commencé par implémenter les types imposés et à faire un plan des fonctions qu'il nous faudra programmer pour créer notre projet. Nous avons ensuite suivi notre plan. Notre première difficulté rencontrée a été de récupérer les 2 entiers entrés en argument par l'utilisateur. Ayant pallié à se problème avec l'utilisation de Sys.argv (qui renvoie les paramètres entrés par l'utilisateur sous forme de tableau de string) et Array.get (qui permet de récupérer la valeur d'une case d'un tableau) nous avons continué notre programme. Nous avons rencontré d'autres problèmes sur certaines fonctions comme la fonction d'affichage graphique pour supprimer les murs au niveau de la reconnaissance des "side" lors de filtrages avec les match d'où la création des fonctions « chartoside » et « sidetochar ». Nous avons ensuite dû comprendre le fonctionnement du type imposé "border" qui permet de stocker les cellules dont au moins une des cellules voisines est fermée. Après avoir créé les fonctions relatives à la gestion du type border. Nous avons été confronté à un nouveau problème pour le choix aléatoire du coté de la première case du labyrinthe à supprimer car la fonction random est toujours initialisée de la même manière et donne donc toujours les même coordonées et le même coté pour la cellule de départ pour la construction du labyrinthe. Nous avons donc décidé que la fonction « Random.init » n'étant pas vraiment « aléatoire », nous allions employer « Random.self_init() » qui est capable de générer des random aléatoires. Nos fonctions d'ajout d'une cellule et celle de suppression d'une cellule dans la bordure ont été sources de problèmes elles aussi dans l'éxécution de notre programme principal car elles crééaient des boucles infinies. Après avoir mis toutes nos fonctions bout à bout nous avions terminé la génération et l'affichage du labyrinthe. Nous commençames donc le calcul et l'affichage du parcours, ce qui ne nous posa pas de gros problèmes. Notre programme était donc terminé.

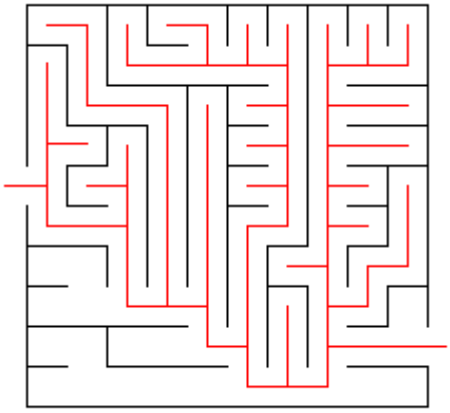
Manuel d'utilisation:

Pour créer le .exe pour windows il suffit de double cliquer sur le fichier creer.bat ensuite il faut lancer le programme via l'invite de commande ms-dos avec en 1er paramètre la largeur, en 2ème la hauteur et en 3ème la vitesse (entre 1 et 10 (1 rapide, 10 très lent)),

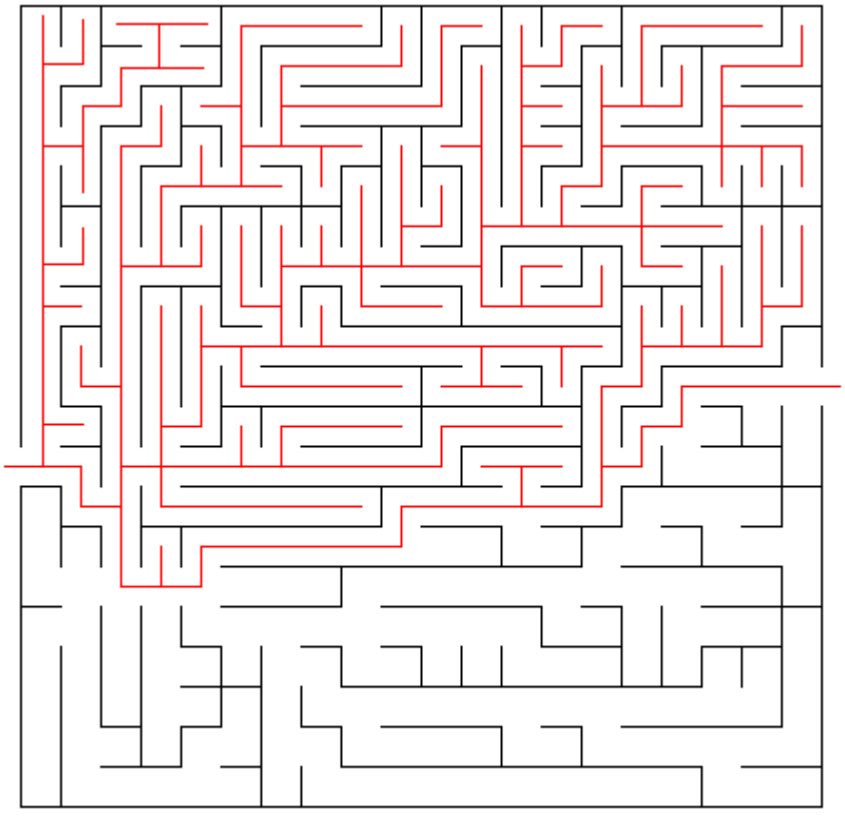
Exemples d'éxécution:

Voici des exemples d'éxécution:

labyrinthe 10 10 5 :



labyrinthe 20 20 5 :



Code:

Le code est fourni avec le texte.

Conclusion:

Ce projet nous a permis d'apprendre à travailler en binôme et à gérer notre travail en fonction du temps imparti, cependant par faute de temps nous n'avons pu apporter des améliorations significatives à notre programme qui comporte encore quelques anomalies notamment une à l'éxécution.